Harikrishnan Suresh

Team C: Fly Sense

Teammates: Shivang Baveja, Nicholas Crispie, Joao Fonseca, Sai Nihar Tadichetty

ILR03

Oct. 27, 2017

1. Individual progress

1.1 Optimization of octomap

I continued working on the 3D mapping ROS implementation, mainly optimizing the octomap algorithm. The focus was to tackle some of the challenges mentioned in the last report, specifically the problem of real time 3D mapping of moving obstacles. Having read the paper "Hornung, Armin, et al. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees." *Autonomous Robots* 34.3 (2013): 189-206" based on which the code was developed, I got an idea that the main parameters to be modified were the clamping thresholds for dynamic updates, the sensor hit and miss probability, and the octree node resolution. The default parameters were suited to their testing conditions which included static obstacles only. After repeated trial and experimentation on our sample data, the parameters were modified to suit a relatively more dynamic environment. The impact of the modified parameters is explained below:

- Reducing the resolution of the octree from 0.05 to 0.1 improves computation speed, and caused faster updates.
- Reducing the minimum probability for clamping from 12% to 2% improves the update frequency. The maximum probability remains the same. So, essentially the algorithm does not wait for a cell to change from occupied to free or vice versa for any specific number of time steps before updating, but updates it close to real time.
- Modifying the sensor model hit rate (70% to 90%) and the miss rate (40% to 20%) causes the octomap algorithm to trust the sensor more, and hence allows dynamic changes in sensor readings.

The parameters were modified using the rqt_reconfigure node in ROS. Figure 1 shows the performance of octomap on the dynamic environment before and after making the changes in values of the parameter. However, the 2D projected obstacle map was not updated in real time and shows up in a very untidy manner. Hence it cannot be used as a 2D map for bird's eye view. The octomap parameters were also modified to remove the unnecessary markers on the floor by changing the minimum height to be considered for mapping. The sensor range was also extended from 5m to 20m. It is seen that as the distance from the sensor increases, the accuracy of 3D mapping decreases.

The working octomap code was run on our onboard computer (Jetson TK1) after Shivang set up the ROS environment on it.

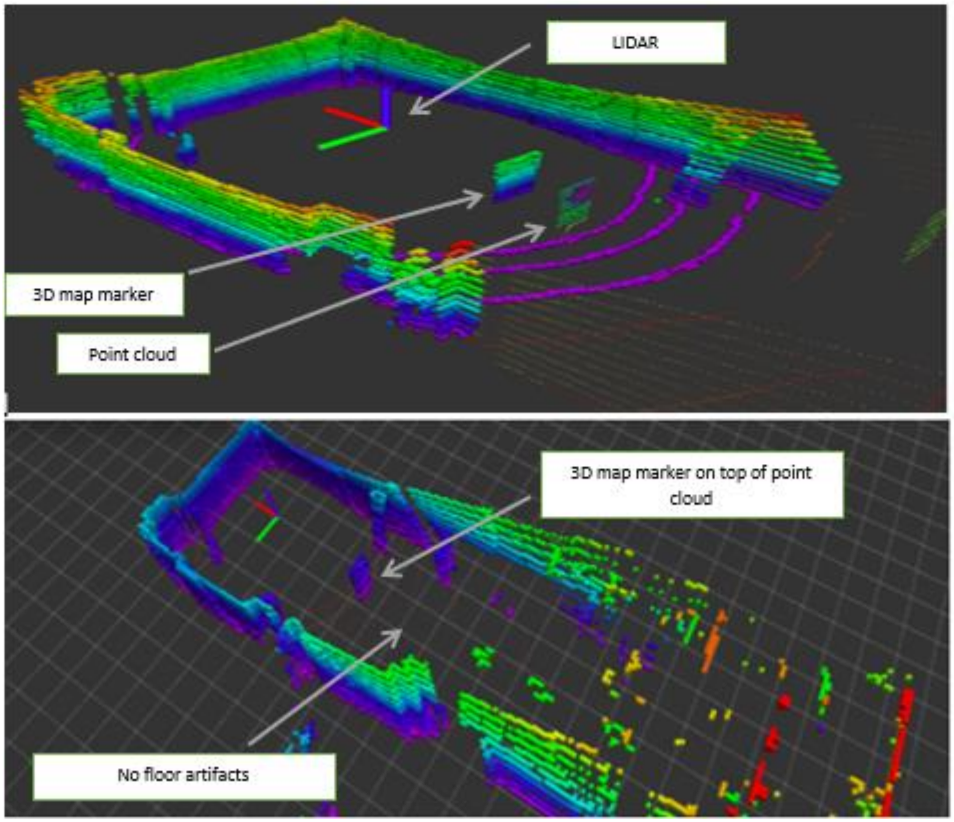


Figure 1: Octomap before (top) and after optimization (bottom)

1.2. Trials with new package - ETHZ grid map

Apart from this, I also started exploring a ROS package grid_map developed at ETH Zurich. (https://github.com/ethz-asl/grid_map). The package had a few demos, mainly visualizations in Rviz. Some of the Rviz plugins seem to be very suitable for our bird's eye view, especially the edge detection filter shown in Figure 2.

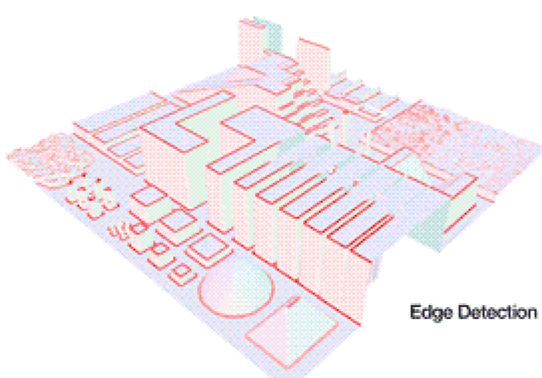
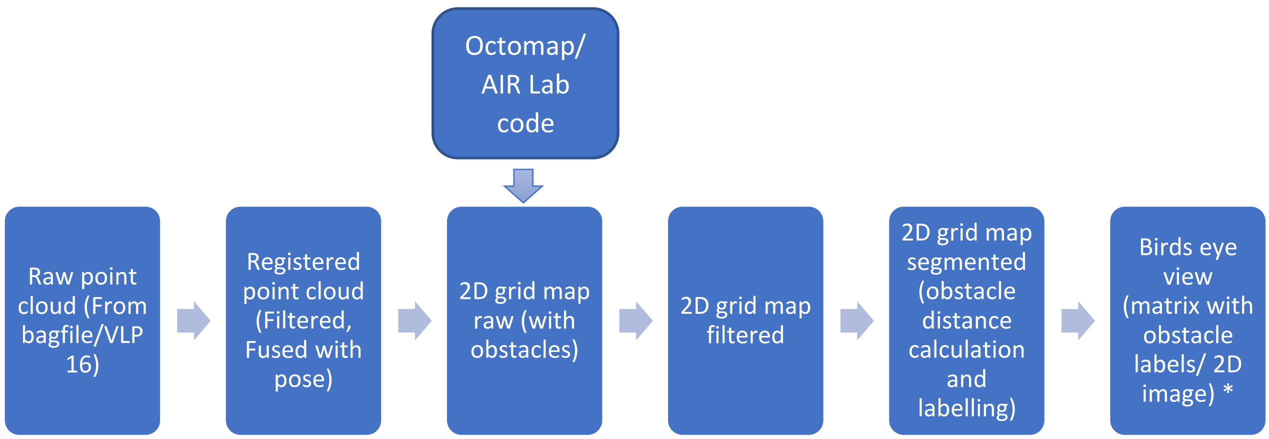


Figure 2: Demo of edge detection filter (from grid map github page)

Looking more into the package, it contains a node that converts and octomap to a grid map in the format the package supports for its other functionalities. It also provides iterators for the grid map as per our convenience. This could be used to specifically identify those nodes that contain obstacles, and in turn compute the distance of the obstacle from the vehicle. Based on the flight dynamics and the calculated distance, the obstacles can be segmented into different categories of danger and accordingly colored.

1.3. Work breakdown structure for the perception system

After reading through some relevant literature and discussions with the team, I have listed a work breakdown structure for the perception system.



* To be decided

1.4. Some investigations on 3D sound

I had a brief discussion with one of my friends (currently Music Technology M.A., McGill university) who also works on 3D sound. He conducted a few tests with beep sounds using one of his tools, and an existing 3D audio tool. From his small experimentation, he concluded that the spatial difference in sound sources can only be felt sounds like music, chirps or human voice and not beeps. Implementing a 3D system for beeps might require a lot of extra effort in research and implementation, and is thus removed from the project after discussions with the team. We now plan to do simple stereo balancing between left and right, and give warnings accordingly.

2. Challenges faced

Some of the challenges faced in my work are listed below:

- The octomap works perfect for the LIDAR data set based on which it was optimized. But testing it with another sample where the obstacle was moving into the frame, the 3D markers kept accumulating all along the path. Following this, I decided to stop optimization and move to 2D obstacle map generation. The optimization will be done after the first version of bird's eye view is developed.

- The octomap to grid map convertor in grid_map package does not run on ROS. The package includes dependencies to some external package that provides an octomap, and probably runs only if it receives an octomap. So, I have started developing a ROS code that subscribes to the octomap_server and generates a grid map in the format supported by the package. The grid map octomap convertor code will be used as reference.

3. Team work

We had several rounds of team discussion for the Systems Engineering presentation, and came up with the least resistance path to our final goal (testing with a helicopter).

We plan to show the complete Flysense system (standard suite, bird's eye View) working in Simulation on the NEA data set, with the user changing modes using voice commands getting visual feedback from the AR device and audio feedback (left and right). We then proceed to demonstrate the same on a live system - sensors in a moving platform and then a quadcopter. After a series of tests with our quadcopter and the NEA quadcopters, we aim to do our final test on a NEA helicopter.

We also had a discussion with Marcel from NEA, who approved to the above schedule and procured a Velodyne PUCK.

Going into the individual contribution of the team members,

Shivang was involved in getting the onboard computer up and running with the complete ROS environment. Once the octomap implementation was completed by me, it was directly fed into the onboard computer and run without any errors.

Nick was involved in the Power Distribution Board design, and in the systems engineering work – WBS schedule, Risk analysis.

Joao had a brief meeting with a Professor in LTI to get some inputs on voice commands and noise cancellation. He also identified an Android offline library that performs speech to text conversion. He has also started reading on the quadcopter dynamics, and flight envelope calculation.

Nihar worked on setting up a basic android application for the Epson BT300. He completed an initial demo with the standard instruments set (showing accelerometer readings) and the bird's eye view. Nihar, in collaboration with Joao also completed an initial draft on the Jetson-AR communication protocols.

4. Plan for next week

My plan for the following week is to develop a ROS code that converts the octomap to a grid map. I will also be considering the 2D obstacle occupancy grid map code developed by the AIR Lab at CMU. We have obtained permission to access the code, and will start working with it once we receive it from our mentor. The two algorithms will be compared and evaluated in different scenarios, and the algorithm most suited for our application will be selected.

Our major plan as a team is to design our fall validation experiment. We also plan to procure various components to conduct tests with a quadcopter.